

PLEASE ENTER TM 05/08/2006

IN THE CLAIMS

1. (Currently Amended) A method comprising:
 identifying scratch values generated during speculative execution of a processor;
 setting ~~at least one~~ tag associated with ~~at least one data area of the processor~~ a register to
 indicate that the ~~data area register~~ holds a scratch value, when an instruction having the register
as a destination results in a cache miss; and
 bypassing execution of instructions having at least one operand with an associated tag
 that indicates that the operand is a scratch value.
2. (Cancelled)
3. (Previously Presented) The method of claim 1 further comprising:
 propagating the tag to destination registers of the instructions to indicate that an operand
 within the destination registers is a scratch value.
4. (Previously Presented) The method of claim 1 further comprising:
 bypassing execution of an arithmetic instruction having at least one register as an operand
 with an associated tag indicating that the register contains data that is a scratch value; and
 bypassing execution of a store instruction involving a value derived from a register
 having an associated tag indicating that the register contains data that is a scratch value.
5. (Original) The method of claim 1 further comprising:
 utilizing a branch predictor to override computed branch results produced by branch
 instructions based on data having a tag that indicates the data is a scratch value.
6. (Original) The method of claim 1 further comprising:
 marking each instruction in a pipeline with a tag to indicate if the instruction involves a
 scratch value.
7. (Original) The method of claim 1 further comprising:
 propagating the tag through a store buffer if an address generation register does not
 indicate that the address generation register holds a scratch value.